Ontology-based database access with DIG-MASTRO and the OBDA Plugin for Protégé

Antonella Poggi[1], Mariano Rodriguez-Muro[2], Marco Ruzzi[1]

[1]Dip. di Informatica e Sistemistica, SAPIENZA University of Rome
[2]Faculty of Computer Science, Free University of Bozen-Bolzano
[1]`lastname@dis.uniroma1.it`, [2]`rodriguez@inf.unibz.it`

Abstract. In Ontology Based Data Access (OBDA), the aim is to use an ontology to mediate access to data. The main contribution of this work is to demonstrate two key components of an OBDA system, whose combination allows ontology practitioners to finally realize end-to-end OBDA systems. The first component is an OBDA-enabled reasoner, named DIG-MASTRO. The second component is the OBDA Plugin for Protégé, that in conjunction with the ontology editing features of Protégé, functions as an OBDA system designer.

1 Introduction

Recent research and industrial efforts in both Data Integration, Semantic Web, or E-Science, witness increasing needs for semantically driven data access, and in particular for the so called *Ontology Based Data Access (OBDA)*. Abstracting from the specific context, the aim of OBDA is to use an *ontology*, i.e. a formal conceptualization of the application domain, to mediate access to data. The added value of OBDA, w.r.t. accessing a data source directly, is twofold. On the one hand, the ontology provides a semantic account of the application data domain. On the other hand, constraints expressed by the ontology allow to overcome incompleteness that may be present in the actual data.

A lot of work has been invested in the last years to realize OBDA technologies. Efforts have been put in defining the appropriate language for the semantic layer, defining the structure and language of the *mappings* used to link the data and the semantic layer, studying the complexity of offering a set of useful services such as query answering, database schema extraction, inconsistency management, etc. Prototype and industrial level systems have been and are being built within academic and industrial research labs including those of major database players such as IBM [4]. However, as pointed out in [3] in the context of data integration, the gap between theory and practice is still wide. The systems which have been made available to the eager community of early adopters do not offer a cohesive structure. They are either based on different assumptions, rely on different user interaction mechanism or provide limited functionality. As a result, when brought together, sometimes in very rough ways, their interaction turns out to be poor.

The main contribution of this work is to present two key components of an OBDA system, whose combination allows ontology practitioners to finally realize end-to-end OBDA systems. The first component is an OBDA-enabled reasoner, named *DIG*-MASTRO . The second component is the *OBDA plugin* for the standard ontology editing tool Protégé [1], that in conjunction with Protégé, functions as an OBDA system designer.

[1] `http://protege.stanford.edu/`

2 From DIG-compliant to OBDA-enabled reasoner: DIG-MASTRO

The *DIG interface* [2] is a standardized HTTP/XML interface to Description Logics (DL) reasoners, that was developed by the DL Implementation Group, with the aim to ease the communication among tools making use of DL reasoners [3]. In particular, by implementing the DIG interface, DIG-compliant reasoners provide standard reasoning services over DL ontologies, e.g. ontology consistency checking, and query answering [1].

Similarly, OBDA-enabled reasoners are systems that implement the OBDA extension to the DIG interface proposed in [2]. By doing so, OBDA-enabled reasoners provide standard reasoning services over ontologies whose data layer is specified by means of *mappings*, i.e. a set of assertions that establish the relationships between the ontology elements and the data at any suitable, possibly autonomous, source.

As an example of OBDA-enabled reasoner, we demonstrate DIG-MASTRO. MASTRO is the realization of the research done over the *DL-Lite*$_{\mathcal{A}}$ language [5]. *DL-Lite*$_{\mathcal{A}}$ is a fragment of OWL-DL designed to maximize expressivity while keeping reasoning algorithms tractable. Specifically, on one hand, *DL-Lite*$_{\mathcal{A}}$ is expressive enough to capture most of the standard UML and ER language constructs. On the other hand, reasoning in *DL-Lite*$_{\mathcal{A}}$ is LOGSPACE in data complexity, as efficient as query answering in relational databases. Therefore, instance level reasoning in *DL-Lite*$_{\mathcal{A}}$ is orders of magnitude lower than reasoning in other DLs.

As far as we are aware of, MASTRO is the only DL reasoner that, besides implementing traditional reasoning services, is specifically built for OBDA operations, incorporating facilities to specify data sources and mappings, and taking these into account when reasoning. Notably, the mapping techniques devised for MASTRO also address the impedance mismatch issue that exists between the values stored in relational data sources and the objects represented in the ontologies [5].

DIG-MASTRO enables MASTRO to interact with any client conforming to the OBDA extension of DIG. Moreover it extends MASTRO querying facilities by offering a set of additional services (the ones defined by the DIG Interface) which are reduced to the answering of suitable Union of Conjunctive Queries (UCQs) that MASTRO can answer. Given the low computational complexity of MASTRO, we argue that DIG-MASTRO is an attractive component of an OBDA system, where the amount of data stored in the sources is large, and efficient reasoning is mandatory.

3 OBDA Plugin for Protégé

On the road to OBDA, one of the main issues is the lack of tools for the design of core components of an OBDA system. With the aim of contributing to this issue, we developed the OBDA Plugin for Protégé, allowing users to: (*i*) describe the data sources of the OBDA system; (*ii*) describe the mappings connecting the data source and the entities of the ontology; (*iii*) send the descriptions of these components to an OBDA-enabled reasoner; and, (*iv*) issue queries to an OBDA-enabled reasoner and view the results.

[2] `http://dl.kr.org/dig/`

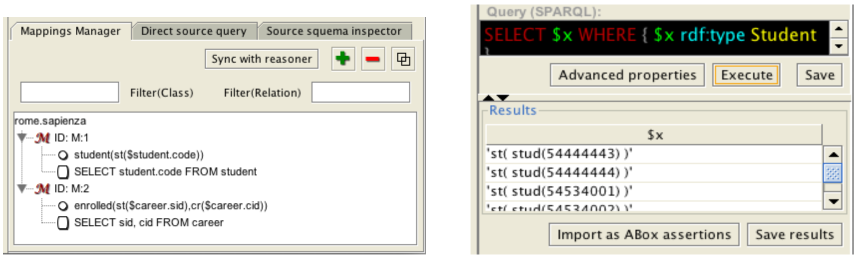
[3] `http://www.cs.man.ac.uk/~sattler/reasoners.html`

Moreover, with flexibility as an objective, we envision an add-ons architecture for the OBDA Plugin for Protégé, to enable third parties to incorporate new suitable features to OBDA systems. Add-ons may either be specific to an OBDA setting, e.g. to enable the plugin to handle new kinds of mappings, data sources or OBDA queries. Or, add-ons may provide generic OBDA functionalities, e.g. to facilitate the design process, data source inspection, query or mapping validation, etc.

In the current implementation, we provide facilities to (*i*) describe relational DBMS data sources; (*ii*) describe mappings of the form $\psi \rightsquigarrow \phi$, where ϕ is a conjunctive query over the ontology, and ψ is an arbitrary SQL query over the data sources [5]; (*iii*) issue UCQs in a restricted SPARQL syntax to the OBDA-enabled reasoner and view the results; and, (*iv*) inspect and manipulate the relational sources.

4 Demonstration scenario

In this section we illustrate the scenario that we will use, to demonstrate how the combination of DIG-MASTRO and the OBDA Plugin for Protégé, together with Protégé, allows to realize and query an OBDA system. Specifically, we will consider the academic domain benchmark ontology LUBM[4], and show how it can be used, within Protégé, to access a database from the Sapienza University of Rome. To emphasize the efficiency of OBDA instance level reasoning and query answering using DIG-MASTRO, we will exploit a very large database, storing information about courses, exams, students and administrative staff within 27 different tables, whose size ranges from few to hundreds of thousands of tuples, with an overall size of 250.000 tuples.



```
rome.sapienza
▼ ID: M:1
    ○ student(st($student.code))
    □ SELECT student.code FROM student
▼ ID: M:2
    ○ enrolled(st($career.sid),cr($career.cid))
    □ SELECT sid, cid FROM career
```

```
SELECT $x WHERE { $x rdf:type Student
}
```

$x
'st(stud(54444443))'
'st(stud(54444444))'
'st(stud(54534001))'
'st(stud(54534002))'

(a) RDMS-Ontology Mapping Pane (b) SPARQL UCQs Pane

Fig. 1. OBDA Plugin screenshots

We next enumerate the key functionalities of DIG-MASTRO and the OBDA Plugin for Protégé that we will demonstrate under the above mentioned scenario.

Ontology definition Since the LUBM ontology is written in the OWL-DL language, and DIG-MASTRO works with ontologies expressed in *DL-Lite*$_{\mathcal{A}}$ (i.e. a sub-fragment of OWL-DL), when the OBDA Plugin for Protégé "tells" DIG-MASTRO about LUBM, it possibly rewrites non-*DL-Lite*$_{\mathcal{A}}$ axioms into equivalent *DL-Lite*$_{\mathcal{A}}$ axioms, and discard those axioms that are beyond *DL-Lite*$_{\mathcal{A}}$, issuing a warning whenever this happens. Remarkably, only few axioms of LUBM are beyond *DL-Lite*$_{\mathcal{A}}$.

[4] http://swat.cse.lehigh.edu/projects/lubm/

Data source and mappings specification Using the OBDA Plugin for Protégé, the user can specify a relational database as a data source, as well as a set of mappings between the data and the ontology. The syntax of the mappings in the OBDA plugin is of the form presented in Section 3. The semantics depends on the OBDA-enabled reasoner, that is DIG-MASTRO in our scenario. Hence, informally, a mapping of the form $\psi \leadsto \phi$, states that for each data tuple $\bar{t}$ satisfying ψ, there exists a set of objects that are built starting from $\bar{t}$ by means of suitable (Skolem) functors, and satisfy ϕ over the ontology [5]. Figure 1(a) shows a screenshot of the plugin *RDMS-Ontology Mapping Pane*, allowing to specify two mappings M : 1 and M : 2.

Query answering Query answering capabilities are provided by the OBDA Plugin for Protégé and its interaction with an OBDA-enabled reasoner. At a glance, the user accesses the plugin *SPARQL UCQs pane*, shown in Figure 1(b), to issue UCQs to an OBDA-enabled reasoner. Hence, as before, the semantics of query answering depends on the OBDA-enabled reasoner, that is DIG-MASTRO [5]. Informally, taking into account the ontology, the data source and the mappings, the reasoner implements a rewriting technique which translates the input UCQ into a set of queries over the source, whose union stands for the answer to the original query. Finally, the plugin displays the results, and allows the user to further manipulate them, e.g. by exporting, or saving them.

5 Conclusion

We presented two contributions in the direction of realizing OBDA systems: (*i*) an OBDA-enabled reasoner named DIG-MASTRO, and (*ii*) a plug-in that enables Protégé to act as an OBDA system designer. We described a scenario in which we use the OBDA Plugin for Protégé to set up and query an OBDA system based on DIG-MASTRO. We now conclude by enumerating some of the features of both components that will be additionally demonstrated during the demo session: (*i*) standard reasoning services, other than query answering, provided by an OBDA system based on DIG-MASTRO, through the use of the OBDA Plugin for Protégé; (*ii*) the workflow of an OBDA system design using the OBDA Plugin for Protégé; and, (*iii*) the deployment of simple OBDA web applications that access an OBDA system based on DIG-MASTRO.

References

1. S. Bechhofer. The DIG Description Logic Interface: DIG/1.1. *Proceedings of the 2003 Description Logic Workshop (DL 2003)*, 2003.
2. D. Calvanese and M. Rodriguez. An extension of dig 2.0 for handling bulk data. In *Proc. of the 3rd Workshop on OWL: Experiences and Directions (OWLED 2007)*, volume 258, 2007.
3. L. M. Haas. Beauty and the beast: The theory and practice of information integration. In T. Schwentick and D. Suciu, editors, *ICDT*, volume 4353 of *Lecture Notes in Computer Science*, pages 28–43. Springer, 2007.
4. L. Ma, J. Mei, Y. Pan, K. Kulkarni, A. Fokoue, and A. Ranganathan. Semantic web technologies and data management. In *Proc. of W3C Workshop on RDF Access to Relational Databases*, 2007.
5. A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. on Data Semantics*, X:133–173, 2008.

[5] Refer to [5] for a full specification of mappings in MASTRO.